FIG. 3 illustrates a cache addressing scheme that permits a portion of the cache 310 to be allocated to one or more tasks in accordance with the present invention. The exemplary cache structure and main memory addressing of FIG. 3 performs in a similar manner to the corresponding conventional functions shown in FIG. 1. The allocation scheme of the present invention introduces a mapper 320 and a map register 330 to the conventional cache addressing scheme of FIG. 1.

As discussed further below, the mapper 320 transforms the set index, A, portion of the memory address to a mapped set index, a, using a set of map bits, M, in accordance with a mapping function. The mapped set index, a, constrains a given task to the corresponding allocated section 140 of the cache. Thus, if the set index, A, identifies a given set 0 to N, the mapped set index, a, identifies a given set in the constrained allocated cache section 140. The allocated cache section 140 of the cache 310 can be varied by selecting an appropriate map function to be implemented by the mapper 320. The map register 330 identifies the allocated section 140 for each task.

In one exemplary embodiment, the mapper 320 may be implemented as a logical bit-wise AND function, allowing the following useful maps:

$M_i = 1$, for all i: entire cache is allocated;

$M_i = 0$, for all i: only set 0 is allocated;

$M_i = 1$, for all i except $i=n-1$, (n is width of set index): lower half of cache is allocated; and

$M_i = 1$, for $i < n-3$: lower eighth of cache is allocated.

In a further variation discussed below in conjunction with FIG. 5, the mapper 320 may be implemented as a multiplexer allowing additional flexibility to isolate allocated sections 140 of the cache. FIG. 4 illustrates the map function, M, between the set index, A, and the mapped set index, a, for the exemplary logical bit-wise AND function.

The tag bits (bits 15 through 31 in the exemplary conventional cache addressing scheme of FIG. 1) must be increased to correspond to the smallest cache section (i.e., by the number of bits set to zero in M). Consequently, as the allocated section gets smaller, the tags become larger. A reasonable compromise may be achieved by limiting the smallest allocated section size to 1/32nd of the cache size. This corresponds to five (5) extra bits for the tag.

Furthermore, in this embodiment, the bits that are set to zero must be contiguous from $M_{n-1.}$ Thus, the cache is allocated from set 0 and up. With limited size of the smallest section, the map register 330 requires only the bits corresponding to the fraction of the cache that the smallest section comprises. For example, for allocating a cache into eight portions, only three bits are required for the map register 330.

Selectable Allocated Cache Section

As previously indicated, the exemplary embodiment discussed above in conjunction with FIGS. 1, 3 and 4 must allocate the cache consecutively beginning with set 0. FIG. 5 illustrates a cache addressing scheme that permits a portion of the cache 310 to be selectively allocated to one or more tasks in accordance with another embodiment of the present invention. With selective allocation, a cache having a smallest allocable section of s sets can be allocated in sections of size N/s, where s is some fraction of the N sets in the cache. The map function, M, required for selective allocation is shown in FIG. 5. The size allocation and selection logic in the address scheme of FIG. 5 requires a 2-to-1 multiplexer 530-i for each address line, i, affected.

A map register 520 stores the size of a section and a map register 510 stores the section selection. The size select bits, M, from the size map register 520 determine the size of the segment. For example, if the minimum cache section is an eighth of the overall cache size, three (3) bits are required (M[2:0]) to select the size of the section. The selection also requires three (3) bits (P[2:0]) to select the section that is allocated. As shown in FIG. 5, an M bit set to a binary value of one (1) allows the set index to be used directly. If an M bit is set to a binary value of zero (0), then the corresponding P bit is selected by the multiplexer 530. For example, for M having a value of "000" and P having a value of "010," the third eighth of the cache is allocated. Similarly, for M having a value of "001" and P having a value of "01x," the second quarter of the cache is allocated.

The selective allocation scheme shown in FIG. 5 is useful, for example, for allocating sections to nested secondary tasks. In summary, the embodiment of FIG. 5 allows a cache to be allocated in sections, with section size and section location being selected by writing a map register. The logic requires only a 2-to-1 multiplexer 530 in the critical address path, and a small number of bits for each tag.

It is to be understood that the embodiments and variations shown and described herein are merely illustrative of the principles of this invention and that various modifications may be implemented by those skilled in the art without departing from the scope and spirit of the invention.